

論文2002-39SD-8-9

X-색도 좌표를 이용한 색온도 변환 시스템 구현

(Implementation of Color-Temperature Conversion System
using X-Chromaticity Coordinate)

李鎬男*, 李奉瑾*, 文五學**, 姜鳳淳**

(Honam Lee, Bonggeun Lee, Ohak Moon, and Bongsoon Kang)

요 약

본 논문에서는 1차원 색도 좌표를 이용한 색온도 변환 방법을 제안한다. 또한, 제안한 알고리즘에 대한 하드웨어 설계 및 구현을 실현한다. 제안한 방법의 성능을 검증하기 위하여, Robertson 알고리즘에서 제안한 2차원 색도 좌표를 이용한 색온도 측정 결과와 비교를 한다. 본 논문에서 제안한 방법은 색온도 3000OK에서 25000OK범위에서 FPGA를 사용하여 실시간으로 검증을 하였으며, 사용된 device는 Xilinx Virtex FPGA XCV2000E-6BG560이다.

Abstract

In this paper, we propose the color-temperature conversion method using an one-dimensional illuminant chromaticity. It also presents the design and the implementation of the proposed method. The performance of the method is compared with that of two-dimensional conversion method in Robertson's algorithm based on calculated color temperatures. The proposed method is demonstrated experimentally for color temperatures in the range of 3,000OK to 25,000OK with the Xilinx Virtex-E FPGA XCV2000E-6BG560.

I. 서 론

사람이 정경을 바라볼 때, 조명의 영향으로 느껴지는 색조를 인지 광원이라고 한다.^[1-3] 백열등에서는 전체적으로 붉은 색조를 느끼며, 일광에서는 백열등에 비하여 푸른 색조를 느끼는 것을 말한다. 지금까지의 칼라 변환 시스템은 RGB의 변환기술(예, saturation, contrast

등)을 통하여 칼라를 변환하는데,^[4] 이 변환 기술은 입력 영상에서 느껴지는 색조를 고려하지 않고 특정 값으로 RGB를 고정함으로써 칼라의 왜곡현상과 영상의 부자연스러움을 가져온다. 또한, 동일한 영상이라고 하더라도 원 영상의 특징을 고려하지 않고 디스플레이 매체의 특성에 따른 칼라 왜곡보정(Gamma Correction)을 적용하고 있다.^[5] 따라서, 이러한 문제점을 보완하고 사용자가 원하는 다양한 색조로 변환할 수 있는 변환 기술이 필요하다고 할 수 있다.

색온도는 광원의 빛을 수치적으로 표시한 것이데, 이상적인 완전흑체가 방출하는 빛의 색은 플랑크의 복사 법칙에 의해 온도가 정해진다.^[6] 물체가 가시광선을 내며 빛나고 있을 때 그 색이 어떤 온도의 흑체가 복사하는 색과 같이 보일 경우, 그 흑체의 온도와 물체의 온도가 같다고 보고 그 온도를 물체의 색온도라고 한다. 즉, 물체의 색온도는 같은 색광의 흑체의 온도(단

* 正會員, 三星電子 DM總括 VD事業部

(VD Division, DM Business, Samsung Electronics)

** 正會員, 東亞大學校 電氣電子 컴퓨터工學部

(School of Electrical, Electronic, and Computer Eng., Dong-A Univ.)

※ 본 논문은 2002학년도 정보통신부 IT학과 장비지원

사업의 동아대학교 대응자금에 의해 연구되었음.

接受日字:2001年9月10日, 수정완료일:2002年6月25日

위:OK)로 표시된다. 가령, 전구의 빛은 2,800OK, 형광 등의 빛은 4,500~6,500OK, 정오의 태양빛은 5,400OK, 흐린 날의 낮빛은 6500~7000OK, 맑은 날의 푸른 하늘 빛은 12,000~18,000OK 정도의 색온도이다. 색온도에 대한 연구는 입력 영상에 대하여 사용자가 원하는 칼라의 색온도 (ex: 일광 효과, 석양 효과 등등)를 반영함으로써 다양한 칼라 변환을 추구하는 것을 말한다.^[7] 칼라 색온도는 인간의 감성에 많은 영향을 준다. 색온도가 낮은 붉은 색조의 영상은 사람에게 따뜻한 느낌을 갖게 하고, 색온도가 높은 푸른 색조의 영상은 시원한 느낌을 주지만, 사람마다 느끼는 감성적 만족도는 제각각 달라 특정 색온도로 고정하면 충족도를 떨어뜨릴 수 있다. 색온도에 대한 표시 방법이 Robertson 알고리즘에서 정량화 되어 있는데, CIE-XYZ 색도 좌표계에서는 XYZ의 3차원 정보로 표시하는 것은 매우 복잡하여 XY의 2차원 정보를 이용하여 수치화하였다.^[8] 이 2차원에 대한 정보는 복잡한 수식으로 표시되어 있어 많은 용량의 하드웨어가 요구되기에 구현의 실용성이 문제가 된다고 할 수 있다.

이에, 본 논문에서는 입력 영상의 X 색도 좌표만을 계산하고 이를 이용하여 입력 영상의 색온도를 계산 및 변환하는 방법과 하드웨어 구조를 제안하고자 한다. 입력 영상의 색온도와 사용자가 원하는 색조의 출력 색온도와의 상관 관계를 이용하여 영상의 색조를 변환을 한다. 색온도 계산시에 보다 정확한 색온도 계산을 위하여 자체 발광 영역은 다른 영역의 색조에 많은 영향을 주기에 색온도 계산에서 제외한다.^[1] 자체 발광 영역이란, 발광체 그 자체이거나 영상에서의 반사광을 의미한다.

본 논문의 II장에서는 색도를 계산하는 원리를 간단히 살펴보고, Robertson 알고리즘의 2차원 색도 좌표를 이용한 방법과 다른 1차원 색도 좌표를 이용한 색온도 변환 방법을 제안한다. III장에서는 1차원 색도 좌표를 이용한 색온도 변환 방법의 블록 구성도와 각 블록의 하드웨어 구조 대하여 다루고자 하며, IV장에서는 시뮬레이션 결과 비교와 FPGA로 구현한 하드웨어 검증을 논하면서 본 연구에서 제안한 방법의 타당성을 다루고자 한다.

II. 알고리즘

색조 변환은 영상의 색도 좌표를 이용한 색온도 변

환을 통하여 수행할 수 있는데, 색온도를 이용한 색조 변환에 있어서 영상의 색온도 측정 방법과 이 방법의 하드웨어 구현 가능성이 본 연구의 핵심이다. 따라서, 하드웨어 구현이 용이한 1차원 색도 좌표를 이용한 색온도 측정 방법을 고찰해 보고 Robertson 알고리즘의 2차원 색온도 계산 방법과 차이를 살펴본다.

1. 입력 영상의 색도 측정

입력 영상의 색도를 계산하기 위하여 입력 데이터를 CIE-XYZ 색도 좌표계로 변환해야 한다. 일반적인 영상 신호인 RGB 신호가 입력될 경우 좌표 변환 방법은 다음과 같다.^[6]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} M \\ \begin{bmatrix} R \\ G \\ B \end{bmatrix} \end{bmatrix} \quad (1)$$

여기서 M 매트릭스는 지상파 TV 신호 기준에 따라 정의되는데, NTSC와 PAL 기준인 경우의 M 매트릭스는 표 1과 같다.

표 1. 색도 좌표 변환 매트릭스 (RGB to XYZ)

Table 1. Conversion Matrices for Chromaticity Coordinates (RGB to XYZ).

NTSC			PAL		
0.5881	0.1791	0.1832	0.4306	0.3415	0.1784
0.2897	0.6056	0.1047	0.222	0.7067	0.0713
0.0	0.0682	1.0209	0.0202	0.1295	0.9394

입력 영상의 색도를 계산하기 위하여 입력 영상의 각 필드마다 자체 발광 영역을 제외한 유효 픽셀의 색도 XYZ에 대한 평균 $X_{ave}, Y_{ave}, Z_{ave}$ 를 식(2)와 같이 구한다.

$$\begin{aligned} X_{ave} &= \frac{1}{\text{Row} \times \text{Col}} \sum_{i=0}^{\text{Row}-1} \sum_{j=0}^{\text{Col}-1} X(i, j) \\ Y_{ave} &= \frac{1}{\text{Row} \times \text{Col}} \sum_{i=0}^{\text{Row}-1} \sum_{j=0}^{\text{Col}-1} Y(i, j) \\ Z_{ave} &= \frac{1}{\text{Row} \times \text{Col}} \sum_{i=0}^{\text{Row}-1} \sum_{j=0}^{\text{Col}-1} Z(i, j) \end{aligned} \quad (2)$$

여기서 열(Row)은 수직 방향으로 입력 영상의 총 라인 수를 나타내고 행(Col)은 수평 방향으로의 총 픽셀 수를 나타내며, i 는 i 번째 라인 수이고 j 는 j 번째 픽

셀 수를 나타낸다. 그리고 XYZ의 평균값을 정규화 방법을 사용하여 식(3)과 같이 입력 영상의 대표 색도 좌표값 (x_s, y_s) 을 측정한다.

$$x_s = \frac{X_{ave}}{X_{ave} + Y_{ave} + Z_{ave}} \quad (3)$$

$$y_s = \frac{Y_{ave}}{X_{ave} + Y_{ave} + Z_{ave}}$$

2. 색온도 측정

Robertson 알고리즘의 색온도 측정 방법을 알아보자. Robertson 알고리즘은 1,677°K에서 무한대까지의 범위에서 31개의 isothermure 라인을 규정하고 있는데, 색온도 계산 방법은 XYZ값 중에서 식(3)에 나타난 것과 같이 XY의 2차원 색도값을 이용하여 색온도를 계산한다.^[8] 이 방법은 31개의 isothermure 라인과 입력 영상의 색도값을 비교하여, 현재 색도 값과 인접한 두 isothermure 라인과 상관을계를 이용한 보간법에 의해 색온도를 구한다. 예를 들어 입력된 영상의 색도 좌표가 (x_s, y_s) 이고 i번째의 isothermure 라인 좌표가 (x_i, y_i) 라고 하자. 그리고 라인의 기울기가 t_i 라고 하면 입력 영상의 색도 좌표와 규정된 라인의 거리 d_i 는 다음의 식 (4)와 같이 구할 수 있다.

$$d_i = \frac{(y_s - y_i) - t_i(x_s - x_i)}{\sqrt{1 + t_i^2}} \quad (4)$$

만일 isothermure 라인 i와 인접한 라인을 i+1로 표시하였을 때, 입력 영상의 색온도는 거리의 비를 이용하여 다음과 같이 구한다.

$$T_c = \left[\frac{1}{T_i} + \frac{d_i}{d_i - d_{i+1}} \left(-\frac{1}{T_{i+1}} - \frac{1}{T_i} \right) \right]^{-1} \quad (5)$$

식 (4)-(5)를 이용할 경우, 요구되는 색온도의 정확도에 따라 하드웨어의 bit size가 매우 큰 제공근기, 나눗셈기, 곱셈기, 역비례기 등의 다수의 매크로 블록이 필요하여, 방대한 하드웨어를 사용하는 단점이 있다. 일 예로, 색온도 1°K의 정확도를 보장하기 위해서는 식(4)의 제공근기의 하드웨어 bit size는 입력 주소가 16bit, 각 주소당 출력 값은 최소 20bit 이상 요구되어 제공근기 1개를 하드웨어로 구현하기 위하여 최소 1,310,270bit의 ROM이 필요하다.^[9] 그러므로 Robertson 알고리즘에 의한 색온도 계산 방법은 하드웨어의 용량이 방대하여,

구현할 경우에 하드웨어의 적용 실용성을 떨어뜨린다.

따라서, 본 논문에서는 하드웨어 구현이 상대적으로 용이한 1차원 색도 좌표를 이용한 색온도 측정 방법을 제안하고자 하는데, 색온도 T_c 는 다음과 같이 색도값과 관계가 있다.^[6] 만일 색온도가 4000°K~7000°K 범위에 속한다면 색도 x_s 와 색온도 T_c 와의 관계는 다음과 같다.

$$x_s = -4.6070 \frac{10^9}{T_c^3} + 2.9678 \frac{10^6}{T_c^2} + 0.09911 \frac{10^3}{T_c} + 0.244063 \quad (6)$$

그리고 색온도가 7000°K~25000°K 범위에 속한다면 색도 x_s 는 다음과 같다.

$$x_s = -2.0064 \frac{10^9}{T_c^3} + 1.9018 \frac{10^6}{T_c^2} + 0.24748 \frac{10^3}{T_c} + 0.237040 \quad (7)$$

식 (3)을 이용하여 구한 색도 x_s 를 식(6)-(7)에 대입하면 입력 영상의 색온도 T_c 를 구할 수 있는데, T_c 에 대한 3차 방정식의 근은 매우 복잡한 식으로 유도될 것으로 쉽게 예측할 수 있다. 일반적으로 복잡한 식은 많은 용량의 하드웨어를 필요로 하는데, 본 논문에서는 하드웨어의 복잡도를 줄이기 위하여 하드웨어로 구현하기 용이한 색온도 T_c 를 계산하는 방법을 제안하고자 한다. 식(6)-(7)에 표시된 영역별 색온도의 근사적인 계산 방법을 그림 1에 나타내었다.

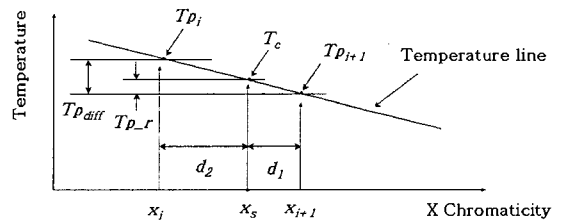


그림 1. X 색도 좌표를 이용한 색온도 측정 방법
Fig. 1. Method of color-temperature calculation using X-chromaticity coordinate.

색도 좌표 계산값은 앞에서 설명한 것과 같이 식(3)을 이용하여 구할 수 있다. 만일 색도 x_s 가 인접한 색도 좌표 x_i 와 x_{i+1} 사이에 놓여 있다면, 색도 간의 거리 d_1 , d_2 는 다음과 같이 구할 수 있다.

$$d_1 = x_{i+1} - x_s \quad (8)$$

$$d_2 = x_s - x_i \quad (9)$$

또한 x_i 와 x_{i+1} 에 해당하는 색온도가 Tp_i 와 Tp_{i+1} 이고 두 색온도 차를 Tp_{diff} 라고 하자. 색온도 차와 색도 간의 거리의 비를 이용하면, 색온도 Tp_{i+1} 와 입력 색온도 T_c 사이의 거리 Tp_r 를 다음과 같이 구할 수 있다.

$$Tp_{diff} = Tp_i - Tp_{i+1} \quad (10)$$

$$Tp_r = Tp_{diff} \times \frac{d_1}{(d_1 + d_2)} \quad (11)$$

마지막으로 Tp_{i+1} 와 Tp_r 를 이용하면 입력 영상의 색온도 T_c 를 구할 수 있다.

$$T_c = Tp_{i+1} + Tp_r \quad (12)$$

따라서, 제안한 방법은 식(4)-(5)에 나타난 복잡한 하드웨어의 용량을 줄일 수 있음을 알 수 있다. 그림 1의 역학구조를 이용하는 색온도 측정을 하여 1개의 나뉜셈기와 곱셈기, 2개의 덧셈기, 3개의 뺄셈기를 사용하면 색온도 계산이 가능하기에 하드웨어의 복잡도를 줄일 수 있다고 할 수 있다.

색온도 변환의 마지막 단계는 색도 정규화 알고리즘을 이용하여 사용자가 원하는 색온도로 변환하면 된다.^[10] 예를 들어, 입력 영상의 색온도가 3000°K이고 사용자가 원하는 출력 색온도가 6500°K이면 색온도 변환 매트릭스 TM 은 다음과 같다.

$$TM = \begin{bmatrix} \frac{X_{white_6500K}}{X_{white_3000K}} & 0 & 0 \\ 0 & \frac{Y_{white_6500K}}{Y_{white_3000K}} & 0 \\ 0 & 0 & \frac{Z_{white_6500K}}{Z_{white_3000K}} \end{bmatrix} \quad (13)$$

여기서 X_{white_3000K} , Y_{white_3000K} , Z_{white_3000K} 는 입력 영상의 색온도가 3000°K일 때의 색도 좌표값을 나타내며, X_{white_6500K} , Y_{white_6500K} , Z_{white_6500K} 는 출력 영상의 색온도가 6500°K일 때의 색도 좌표값을 나타낸다.

위에서 계산된 색온도 변환 매트릭스는 색도 좌표계를 XYZ에서 RGB로 변환하기 위해 아래의 식(14)와 같이 수행된 후에 영상이 화면에 출력되게 된다.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} IM \\ MC \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} MC \\ TM \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (14)$$

여기에서 MC 매트릭스는 표 2와 같으며,^[6] IM 매트릭스는 MC 와 TM 매트릭스를 연산한 결과의 3×3 행렬이다.

표 2. RGB 데이터 변환 매트릭스 (XYZ to RGB)
Table 2. Conversion Matrices for RGB data (XYZ to RGB).

NTSC			PAL		
1.9709	-0.5494	-0.2974	3.0627	-1.3928	-0.4759
-0.9538	1.9364	-0.0274	-0.9689	1.8756	0.0417
0.0637	-0.1294	0.9814	0.0677	-0.2286	1.0690

III. 하드웨어 아키텍처

그림 2는 본 논문에서 제안한 색온도 변환 시스템의 내부 블록도를 나타낸다. 제안한 시스템은 현재 필드의 색온도를 다음 필드에 적용하여 외부에 필드 메모리를 사용하지 않는 하드웨어 구조가 되도록 설계하여 영상의 필드 정보를 저장하는 외부의 메모리가 필요하지 않다. 본 논문에서 제안한 시스템은 10개의 기능 블록으로 구성되어 있는데, (i) CIE-XYZ 색도 좌표 변환부, (ii) 유효 픽셀 선택부, (iii) 색도 계산부, (iv) 필터부, (v) 색온도 계산부, (vi) 변환 매트릭스 생성부, (vii) RGB 변환 매트릭스 생성부, (viii) 원하는 출력 색온도로의 출력 변환부, (ix) 제어신호 생성부와 (x) I2C 레지스터 생성부로 구성되어 있다.

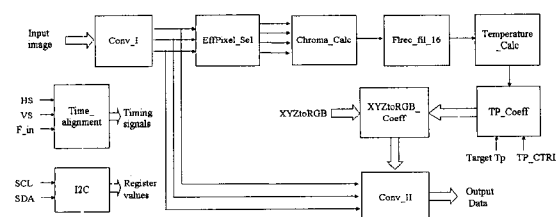


그림 2. 1차원 색도 좌표를 이용한 색온도 변환 장치의 구조

Fig. 2. Block diagram of one-dimensional color-temperature conversion.

Conv_I 블록은 CIE-XYZ 색도 좌표 변환을 위한 좌표 변환부를 나타낸다. 즉, 입력 RGB 데이터를 CIE-XYZ 좌표계의 XYZ 데이터로 식(1)과 표 1을 사용하

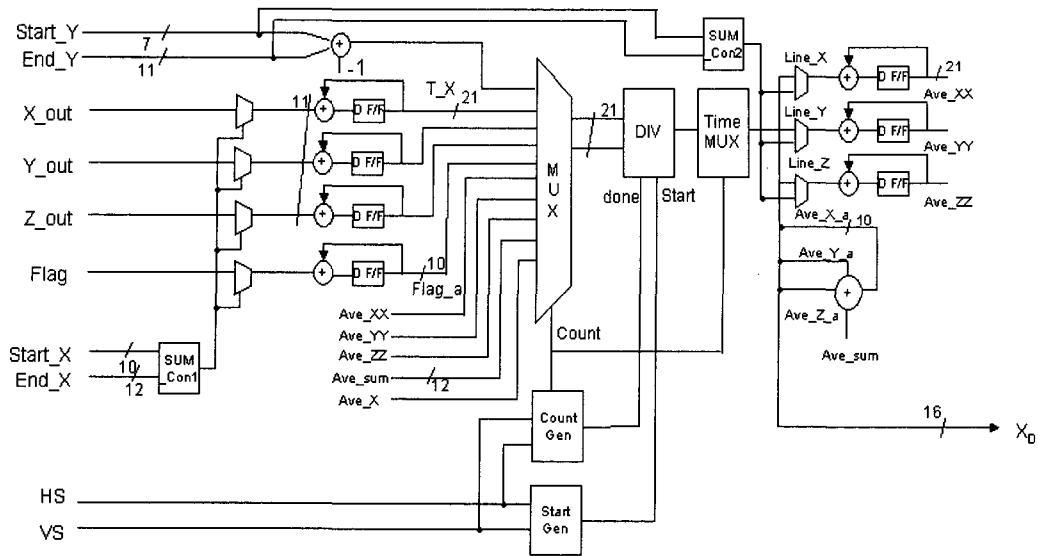


그림 3. 색도 계산부의 구조
Fig. 3. Block diagram of the Chroma_Calc.

여 변환한다. 연산에 사용되는 변환 매트릭스는 영상 신호의 표준인 NTSC, PAL 표준에 따라 색도 좌표 변환 매트릭스의 값이 달라지는데, I2C 블록을 통하여 입력 영상의 표준을 설정한다.

EffPixel_sel 블록은 자체 발광 영역을 제외한 유효 픽셀을 선택하는 유효 픽셀 선택부이다. 변환된 XYZ의 상관관계에 따라 자체 발광체로 인식되는 픽셀을 제외한 유효 픽셀을 판별하는데, 다음 블록에서 사용될 유효 픽셀 flag 신호를 동시에 발생시킨다.

Chroma_calc 블록은 색도 계산부인데, 한 필드내에서 유효한 픽셀의 XYZ 데이터 각각의 평균을 구하고 이를 이용하여 색도를 계산한다. 이에 대한 내용은 식 (2)-(3)에서 설명하였다. 그림 3은 색도 계산부의 상세한 내부 구조이다. 유효 픽셀을 판별하는 flag 신호를 이용하여 XYZ 각각의 평균을 구하고, 입력 영상의 색

도 xs를 계산한다. 그림 3의 색도 계산부에서는 하드웨어의 감소를 위하여 시분할 나눗셈기(DIV) 한 개를 사용하여 입력 영상의 색도값을 구하였다. 이는 색도 계산이 필드에 한번만 계산되면 되기에, XYZ의 각 채널에 대하여 나눗셈기를 따로 구현하지 않고 하나의 나눗셈기를 영상의 정보가 실려 있지 않은 수직동기와 수평동기 시간 동안에 시분할로 연산을 수행하게 함으로써 가능하였다. Start_Y, End_Y, Start_X 및 End_X 신호는 입력 영상의 윈도우 크기 즉 색도 계산의 실제 영역을 알려주기 위한 레지스터 값으로 I2C를 통하여 외부에서 조정되게 구성하였다. 따라서, 색온도 계산을 적용할 영상의 크기는 입력 영상의 유형에 맞추어 조정성이 가능하다.

Flrec_fil_16 블록은 화면의 갑작스러운 변화에 따른 떨림 현상을 제거하기 위하여 저역통과대역 필터를 사

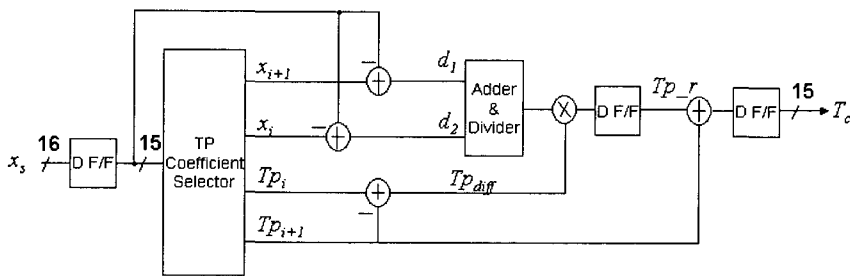


그림 4. 색온도 계산부의 구조
Fig. 4. Block diagram of the Temperature_Calc.

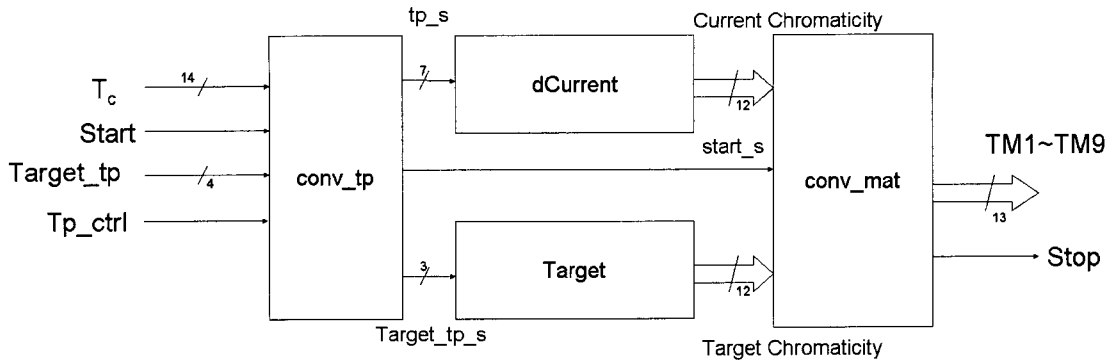


그림 5. 변환 매트릭스 생성부의 구조
Fig. 5. Block diagram of the TP_Coeff.

용하였다. 즉, 영상의 내용이 바뀔 때 색온도 변화의 폭이 크게 되면 떨림 현상이 발생할 수 있으므로 이 경우를 대비하여 내장하였는데 I2C를 통하여 on/off의 제어가 가능하게 설계하였다.

Temperature_calc 블록은 그림 4에서 보는 것과 같이 색온도 계산부로 그림 1에서 설명한 1차원 X 색도 좌표를 이용하여 색온도를 계산한다. xs는 입력 영상의 X 색도값을 나타내며, 이를 이용하여 식(8)-(12)에서와 같이 규정된 색온도 라인을 선택하고 입력의 색온도 Tc를 계산한다.

그림 5는 식(13)에서의, 입력 영상을 원하는 색온도의 출력 영상으로 변환하기 위한 색온도 변환 매트릭스 생성부 구조로써 Tp_Coeff 블록의 상세한 내부 구조를 나타낸다. Tc는 계산된 입력 영상의 색온도이다. 색온도 변환 매트릭스는 하드웨어를 줄이기 위하여 하드웨어를 통하여 직접 계산하지 않고 계산된 값을 ROM table로 정의하였다. 따라서 입력 색온도를 이용하여 conv_tp 블록에서 입력 색온도에 대한 어드레스를 생성하고 이 어드레스를 이용하여 dCurrent 블록에서 입력 영상의 색온도에 대한 매트릭스를 선택한다. Target 블록에서는 사용자가 원하는 색온도 Target_tp 신호를 이용하여 출력 영상의 색온도에 대한 매트릭스를 선택한다. conv_mat 블록은 최종 색온도 변환 매트릭스인 TM을 생성한다. Target_tp 신호는 원하는 출력 색온도를 제어하는 신호를 나타내며, TM1~TM9 데이터는 입력 색온도에 대한 출력 색온도의 변환 매트릭스 값을 나타낸다. 또한 Tp_ctrl 신호는 출력 색온도를 특정 색온도로 고정할 것인지, 입력 영상의 색온도에 따라 가변적으로 출력 색온도를 결정할 것인지를

제어한다.

그림 6은 식(14)에서의 MC 매트릭스와 TM 매트릭스의 곱하여 IM 매트릭스를 생성하는 XYZtoRGB_Coeff 블록으로 XYZ 좌표계 상태에서의 색온도 변환 매트릭스를 영상신호의 표준인 RGB 데이터에 대한 색온도 변환 매트릭스로 변환하기 위한 블록이다. 하드웨어 감소를 위하여 색도 계산부인 Chroma_Calc 블록과 같은 방식으로 시분할 곱셈기를 이용하였다. 색온도 변환 매트릭스는 3x3의 행렬이며, 행렬의 각 원소는 3개의 곱셈기가 필요하다. 따라서 매트릭스 연산에서는 총 27개의 곱셈기가 필요한데 이를 1개의 시분할 곱셈기(⊗)를 사용하여 하드웨어의 용량을 줄였다. MC1~MC9는 XYZ 색도 좌표를 RGB 좌표로 변환하기 위한 매트릭스 데이터이며, TM1~TM9 데이터는 CIE-XYZ 좌표에서의 색온도 변환 매트릭스를 나타낸다. 이 두 매트릭스를 이용하여 사용자가 원하는 최종 색온도에 해당하는 변환 매트릭스(IM1~IM9)를 생성한다.

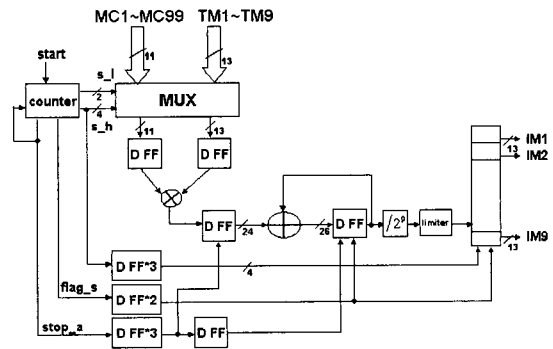


그림 6. RGB 변환 매트릭스 생성부의 구조
Fig. 6. Block diagram of the XYZtoRGB_Coeff.

Conv_II 블록은 식(14)에 의해 입력 영상을 원하는 색온도의 RGB 데이터로 변환하기 위한 3×3 행렬의 출력 변환부인데, 그림 6에서 계산된 변환 매트릭스 IM과 입력 XYZ 데이터를 곱하여 최종 출력인 RGB 데이터를 생성한다.

Time_alignment 블록은 제어 신호 생성부로 각 블록에서 필요한 제어 신호를 발생한다. 즉, 내부의 블록에서 연산을 하기 위해 필요한 각종 수직동기와 수평동기 신호들을 발생한다.

I2C 블록은 입력 영상의 표준(NTSC 혹은 PAL) 설정, 사용자 선호 색온도, 색온도 계산 영상 영역 설정 등 하드웨어 동작과 연관된 레지스터의 값들을 변경하기 위하여 사용되는 프로토콜이다.^[11]

IV 실험 결과

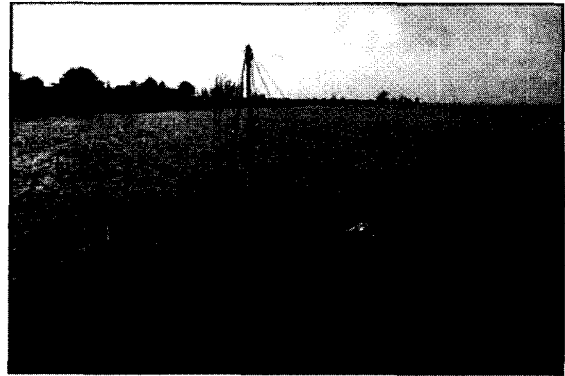
표 3은 본 논문에서 제안한 방법과 Robertson 알고리즘을 이용한 색온도 계산 방법을 사용하여 6개의 서로 다른 입력 영상에 대한 색온도 계산 결과이다. 표 3에 나타나듯이, 본 논문에서 제안한 방법을 사용하였을 때의 결과는 Robertson 알고리즘을 사용하였을 때의 결과와 매우 비슷하였는데, 계산된 색온도 차는 불과 최소 -8°K 에서 최대 34°K 이었다. 이 정도의 오차에서는 사람의 눈으로는 그 차이를 알 수 없었다. 따라서 본 논문에서 제안한 1차원 색도 좌표를 이용한 색온도 계산 방법은 2차원 방법과 비교하여 거의 동등한 수준으로 색온도를 계산하는 것을 알 수 있다.

표 3. 색온도 비교 (단위: OK)

Table 3. Comparison of color temperatures (unit: OK).

Type	Temperature		Difference
	Robertson Method	Proposed Method	
1	4,788	4,780	-8
2	5,182	5,216	34
3	5,278	5,298	20
4	5,773	5,795	22
5	5,942	5,923	-19
6	7,403	7,426	23

그림 7은 본 논문에서 제안한 색온도 변환 방법의



(a) 원 영상 (5,802OK)

(a) Original image (5,802OK).



(b) 변환 영상 (9,500^oK)

(b) Converted image (9,500^oK)



(c) 변환 영상 (3,000^oK)

(c) Converted image (3,000^oK)

그림 7. 색온도 9,500^oK와 3,000OK로 변환

Fig. 7. Color-temperature conversions with 9,500OK and 3,000^oK.

시뮬레이션 결과를 나타낸다. 그림 7(a)는 입력 영상으로 5,802OK의 색온도를 나타내며, 그림 7(b)와 7(c)는

입력 영상을 색온도 9,500°K와 3,000°K로 변환한 출력 영상을 나타낸다. 색온도 9,500°K로 변환된 영상에서는 입력 영상보다 푸른 색조를 나타내며, 색온도 3,000°K로 변환된 영상에서는 입력 영상보다 더 붉은 색조가 나타남을 볼 수 있다. 이는 사람이 더 높은 색온도에서는 푸른 색조를 느끼고, 더 낮은 색온도에서는 붉은 색조를 느끼기 때문이다. 그림 7에서 본 논문에서 제안한 색온도 계산 방법의 타당성을 검증할 수 있다.

본 논문에서 제안한 방법은 VHDL을 이용하여 설계하였으며, Synopsys simulator를 사용하여 검증하였다. 검증된 VHDL 소스 코드의 하드웨어 복잡도를 살펴보기 위하여, 삼성 0.35um STD90 library를 이용하여 합성하였다. 설계된 하드웨어는 ITU-R BT.601^[12]에서 규정한 13.5MHz의 두 배인 27MHz 주파수로 실시간 동작을 하여야 하기에 회로 합성에서 사용된 동작 클럭의 주기는 37ns 이고, 악조건 제약은 전압 3.0V, 공정 worst와 온도 85°C industry 조건인 V300WT0850을 사용하였다. 표 4는 본 논문에서 제안한 1차원 색도 좌표를 이용한 색온도 변환기의 합성된 gate counts와 각 블록에서 critical path을 보여주는 max. timing을 나타낸다. 하드웨어의 복잡도는 일반적으로 2-input nand

gate 기준으로 표시하는데, 본 논문에서 제안한 방법은 97,123 gate가 필요하였다. 또한 설계된 하드웨어의 최대 critical path는 Ffrec_fil_16 블록의 필터 연산부 이었으며, 최대 연산 소요 시간은 24.66ns로 측정되었다. 이를 주파수로 환산하면 40.55MHz로써 하드웨어의 요구 동작 속도인 27MHz를 만족함을 알 수 있다.

제안한 1차원 색도 좌표를 이용한 색온도 변환 방법의 실시간 검증을 위하여 Xilinx Virtex-E FPGA XCV2000E-6BG560을 사용하여 FPGA로 구현하였다. Xilinx FPGA에서 사용된 slice의 크기는 46%였으며, critical path는 표 4의 standard cell과 달리 Tp_Coeff 블록이었으며 최대 연산 소요 시간은 33.22ns로 측정되었다. 이를 주파수로 환산하면 30.10MHz로써 하드웨어의 요구 동작 속도인 27MHz를 만족함을 알 수 있다. Xilinx FPGA의 critical path가 standard cell과 다르게 나온 이유는, Xilinx FPGA에서는 플립플롭보다도 조합 논리회로에서 더 많은 연산 시간이 소요되는 것에 따른 것으로 추정하고 있다. 그림 8은 본 논문에서 제안한 방법을 실시간 검증을 위해 삼성전자의 해당분야 전문가의 협조를 통하여 제작된 검증용 데모 보드이며, 그림 9는 데모 보드를 내장한 29" 완전평면 TV 세트이다. 검증용 데모 보드에서는 외부에서 입력되는 NTSC 등의 아날로그 TV 신호를 비디오 디코더(Micronas사의 VPC3230D-A0)에서 ITU-R BT.601의 디지털 RGB 신호로 변환하고, 변환된 디지털 신호는 본 논문에서 제안된 방법이 구현된 FPGA로 입력되게 구성되어 하였다. 또한, FPGA에서 색온도 변환 과정을 거친 RGB 신호는 비디오 인코더(Infineon사의 SDA9280)에서 아날로그 TV 신호를 생성하여 TV 세트로 입력되게 설계되어 있다. 본 논문에서 제안한 색온도 변환기의 하드웨어 동작에 필요한 레지스터 값들은 PC를 통하여 값들을 변경하게 하였는데, 이 값들은 I2C Protocol를 통하여 제어된다. 그림 9는 본 논문에서 제안한 방법을 실시간으로 비교할 수 있게 하드웨어 내부 동작 레지스터를 설정하였다. 데모 TV 세트 화면의 왼쪽 반은 입력 영상을 그대로 보여주고(bypass mode), 화면의 오른쪽 반은 본 논문에서 제안한 방법을 사용하여 색온도가 12,000°K로 변환된 영상이 실시간으로 나타나게 하였다. 실시간 검증에서도, 입력 영상(왼쪽 반)보다 더 높은 12,000°K로 변환된 영상(오른쪽 반)에서는 푸른 색조가 나타남을 볼 수 있다. 이를 통하여 본 논문에서 제안된 색온도 변환 방법의 타당성

표 4. 1차원 색도 좌표를 이용한 색온도 변환기의 gate counts와 Max. timing
Table 4. Gate counts and maximum timing of one-dimensional color-temperature conversion.

Module Name	Gate Counts	Max. Timing(ns)
Cov_I	9,531	14.63
EffPixel_sel	1,992	9.13
Chroma_Calc	8,020	13.57
Ffrec_fil_16	1,349	24.66
Temperature_Calc	19,150	22.57
Tp_Coeff	34,502	21.39
XYZtoRGB_Coeff	3,883	13.24
Conv_II	12,801	18.30
Time_Alignment	2,611	15.87
I2C	3,284	7.64
Total Gate Counts and Max. Timing	97,123	24.66

및 실용성을 다시 검증하였다.

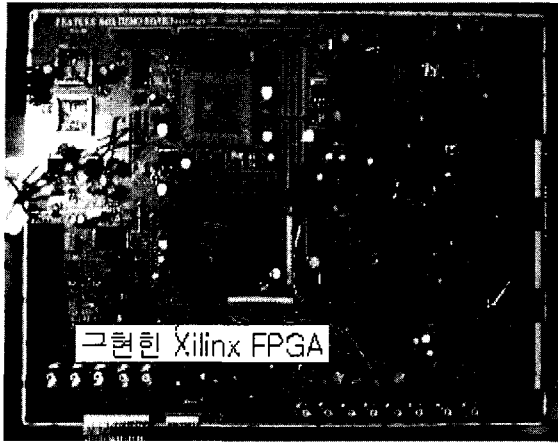


그림 8. 1차원 색도 좌표를 이용한 색온도 변환기의 데모 보드

Fig. 8. Demonstration board of one-dimensional color-temperature conversion.

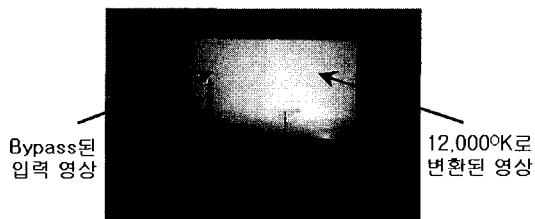


그림 9. 1차원 색도 좌표를 이용한 색온도 변환기의 데모 TV 세트

Fig. 9. Demonstration TV set of one-dimensional color-temperature conversion.

V. 결 론

본 논문에서 제안한 1차원 색도 좌표를 이용한 색온도 변환기는 Robertson 알고리즘의 2차원 색온도 변환 방법과 색온도 계산 결과에서 거의 동일한 수준을 보여주었다. 또한, 하드웨어의 용량을 감소하기 위하여 시분할 곱셈기·나눗셈기를 사용하였으며, 영상의 색온도 계산이 용이한 하드웨어 구조를 채택하여 구현한 하드웨어의 실용성을 최대한 높였다. 실시간 검증을 위하여 Xilinx Virtex-E FPGA XCV 2000E-6BG560를 사용하여 제안한 방법의 타당성을 검증하였는데, TV 세트 실험에서는 3000°K에서 25000°K까지의 색온도 범위에서 색온도 변환을 수행하는 것을 확인하였다. 따라서 본

논문에서 제안한 방법은 색조 변환을 위한 영상 처리 장치인 HDTV, TV, 캠코더, 프린터 등의 다양한 멀티미디어 디스플레이 장치에 응용이 가능함을 확인하였다고 할 수 있다.

감사의 글

본 논문에서 제안한 방법의 타당성을 검증하기 위한 검증용 데모 보드 제작에 아낌없는 협조를 해주신 삼성전자의 최형진선임, 박세웅책임, 홍창완상무님께 감사드립니다. 또한, 본 연구는 IDEC에서 제공한 Synopsys와 Xilinx tool을 사용하여 수행하였습니다.

참 고 문 헌

- [1] A.P. Petrov, C.Y. Kim, Y.S. Seo, and I.S. Kweon, "Perceived illuminant measured, Color Research and Application," Vol. 23, No. 3, pp. 159~168, 1998.
- [2] 강봉순, 장근식, "색온도 변환장치 및 방법", 대한민국 출원 특허, 출원번호: 10-2001-0063418, 2001.1
- [3] 강봉순, "색온도를 이용한 디스플레이 시스템의 칼라 변환 기술", IDEC News Letter, pp. 34~37, 2002.1
- [4] K. Jack, Video Demystified: A Handbook for the Digital Engineer, HighText Pub., 1996.]
- [5] C.A. Poynton, A Technical Introduction to Digital Video, John Wiley & Sons Inc., 1996.
- [6] W. N Sproson, Colour Science in Television and Display Systems, Adam Hilger Ltd., 1983.
- [7] H. Lee, B. Lee, O. Moon, and B. Kang, "Design and Implementation of Color Management System with X-chromaticity Coordinate," 2001-Summer IDEC Conference, pp. 157~158, 2001.
- [8] G. Wyszecki and W.S. Stiles, Color Science Concepts and Methods, Quantitative Data and Formulae, John Wiley & Sons Inc., 1982.
- [9] 강봉순, fLCD용 Magic Color Hardware Implementation, 최종연구보고서, 삼성전자, 2001.6

- [10] 박두식, 색신호 정규화 기법에 대한 연구, 기술보
고서, 삼성종합기술원, 2001.2
- [11] Philips Corporation, Data book, 1999.
- [12] ITU-R BT.601, Encoding Parameters of Digital
Television for Studios, 1994.

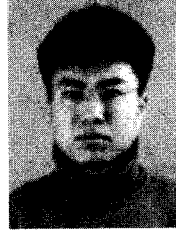
저 자 소 개



李 鎬 男(正會員)

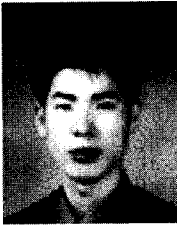
2000년 2월 동아대학교 전자공학과
졸업(공학사). 2002년 2월 동아대학
교 전자공학과 대학원 졸업(공학석
사). 2002년 1월~현재 삼성전자
DM총괄 VD사업부 연구원. <주관
심분야 : ASIC 설계, Digital signal

processing, Image processing 등>



文 五 學(正會員)

2000년 2월 동아대학교 전자공학과
졸업(공학사). 2002년 2월 동아대학
교 전자공학과 대학원 졸업(공학석
사). <주관심분야 : ASIC 설계,
Digital signal processing, Image
processing 등>



李 奉 瑾(正會員)

2000년 2월 동아대학교 전자공학과
졸업(공학사). 2002년 2월 동아대학
교 전자공학과 대학원 졸업(공학석
사). 2002년 1월~현재 삼성전자
DM총괄 VD사업부 연구원. <주관
심분야 : ASIC 설계, Digital signal

processing, Image processing, Wireless commu-
nication 등>

姜 鳳 淳(正會員) 第 37卷 SD編 第 5號 參照